Verificabilidad "End to End" para OTP - Vote

Pablo García[1], Silvia Bast[1], Germán Montejano[1,2]

[1] FCEyN (UNLPam) – Uruguay 151, Santa Rosa, La Pampa
{pablogarcia, silviabast}@exactas.unlpam.edu.ar
[2] FCFMyN (UNSL) – Ejercito de Los Andes 50, San Luis, Argentina
gmonte@unsl.edu.ar

Abstract- OTP – Vote es un modelo para votación electrónica basado en One Time Pad que utiliza claves múltiples y distribuidas que funcionan finalmente como una llave única. El esquema garantiza anonimato incondicional en la medida en que se verifiquen las condiciones iniciales exigidas y, simultáneamente, permite llevar la seguridad computacional del escrutinio a los niveles que se deseen. Como cualquier propuesta relacionada con voto electrónico, uno de los principales inconvenientes a la hora de la implementación pasa por convencer a la sociedad de que el esquema es seguro. Aportando en ese sentido, el presente documento expone una técnica para aplicar verificabilidad "End to End" al esquema original.

Key words: E-Voting – OTP - Vote – Parallel Channels – Verificabilidad "End to End" – Funciones de Hash – MCDU – Birthday Paradox.

1. Introducción

La implementación de sistemas de voto electrónico se encuentra en la actualidad en un momento de transición. Existe una cantidad de argumentos atendibles que dan lugar a que desde muchos sectores[1,2] se afirme que los productos de software diseñados para tal fin no permiten garantizar la transparencia del proceso electoral.

En general, gran parte de la sociedad percibe como una enorme "caja negra" lo que pueda ocurrir con los sufragios una vez que se complete la emisión de los mismos. Si bien no se justifica razonablemente que esa incertidumbre sea mayor que en un sistema de votación manual tradicional, la realidad nos indica que ésa es la percepción dominante. Como consecuencia de lo anterior, todos los aportes que puedan realizarse en búsqueda de garantizar la transparencia del proceso de votación electrónica adquieren relevancia.

[1] http://www.infobae.com/opinion/2016/10/31/problemas-que-el-voto-electronico-trae-y-los-que-no-evita/
[2] http://www.laizquierdadiario.com/Voto-Electronico-un-sistema-vulnerable-que-no-garantiza-el-voto-secreto

En particular, la Verificabilidad "End to End" (E2E, [1], [2], [3]) es uno de los puntos de mayor valor a los efectos de agregar transparencia al proceso de votación electrónica. Se define mediante las tres condiciones siguientes:

- Verificabilidad individual: cualquier votante puede verificar que su sufragio fue incluido en el recuento.
- Verificabilidad universal: cualquier persona puede determinar que el recuento total de los votos es correcto.
- Secreto del voto: ningún votante podrá demostrar cuál fue la opción que eligió, a los efectos de evitar maniobras relacionadas con el "clientelismo político".

2. OTP - Vote

El Modelo OTP-Vote, presentado en [4], se basa en la premisa de que en cualquier sistema de voto electrónico es necesario proteger:

- Indefinidamente la privacidad del votante ([5]).
- La seguridad de los datos del escrutinio mientras dure el proceso electoral. La protección de los sufragios anónimos sólo debe soportar el lapso de tiempo que corresponda al proceso de votación ([6]). Luego, la información se hará pública.

En *OTP – Vote*, el proceso eleccionario consiste de tres grandes etapas:

Figura 1: Etapas del Proceso Eleccionario

La Figura 2 expone las etapas con sus datos de entrada y salida.

Los elementos de datos que aparecen en el Modelo son:

- Claves: el modelo hace uso de claves One Time Pad (OTP, [7]), que está basado en el *Secreto Perfecto* de Shannon [8]: las mencionadas claves son totalmente aleatorias y tan largas como el mensaje claro.
- Archivos de datos que almacenan bits. son elementos fundamentales en el Modelo propuesto y se van modificando durante el proceso eleccionario:
- Clave de Descifrado (*CD*): se genera a partir de operaciones XOR (⊕) de claves OTP.
- Archivo Binario de Votos (ABV) se genera en base al modelo de almacenamiento Múltiples Canales Dato Único (MCDU) propuesto en [9], que se muestra en la Figura 3. El esquema apunta a resolver las limitaciones presentadas por Birthday Paradox [10] y se analiza en profundidad en [11] y [12].
- Tablas Relacionales: se usan para almacenar los datos de los cargos, candidatos e identificadores de votos. También los votos emitidos, una vez finalizado el proceso eleccionario

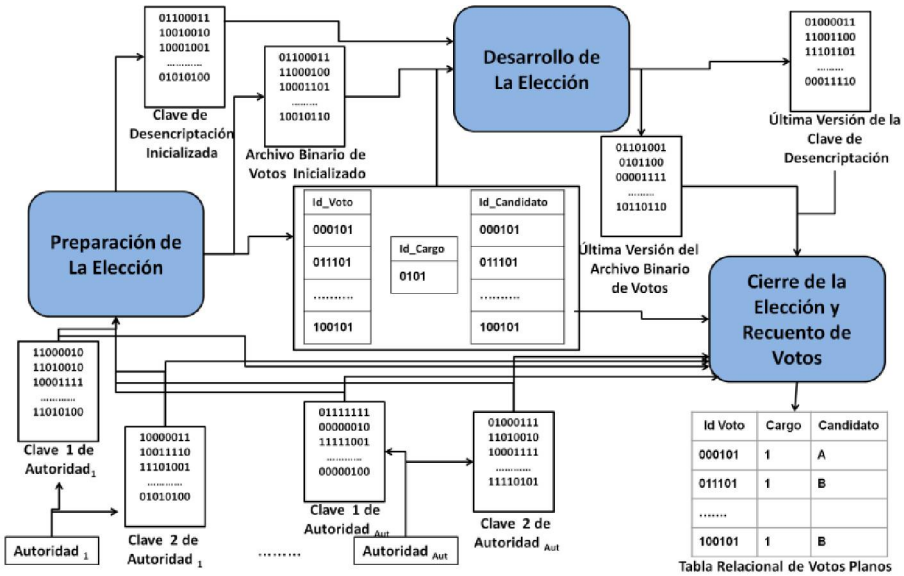

Figura 2: Etapas del Proceso Eleccionario con Datos de Entrada y Salida

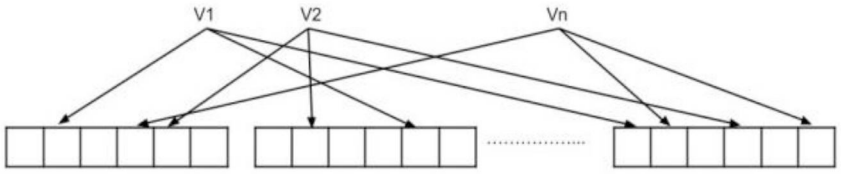

Figura 3: Propuesta de Almacenamiento de *MCDU*

Se describen a continuación las actividades que se llevan a cabo en cada una de las etapas del proceso eleccionario:

Etapa de Preparación de la Elección: las acciones que deben llevarse a cabo en esta etapa son:

- Especificación de la semántica de la tupla en la que se almacenarán los votos:
  - #bits de cada slot (*TBslot*).
  - #bits del Identificador de Voto (*TBId*) y su ubicación.
  - #bits asignados para almacenar el cargo (*TBcargo*) y su ubicación.
  - #bits asignados para almacenar código del candidato votado para cada cargo de la elección (*TBCandidato*) y su ubicación.
- En cuanto a la definición de las dimensiones de los atributos, es necesario evaluar la probabilidad de que algún intruso pueda obtener un dato válido de entre todos los posibles. A través del aumento de la redundancia en la cantidad de bits usados para el almacenamiento de cada uno de los atributos, la probabilidad de obtener una tupla válida de entre todas las combinaciones de valores posibles puede llevarse a cualquier valor deseado.

- Definir las dimensiones del *ABV* y *CD*, según las fórmulas establecidas en [9].
- Generar las tablas: *Cargos*, *Candidatos* e *Identificadores de Votos*, con las características propuestas en [4]
- En el momento previo al comienzo del acto eleccionario se inicializan el *ABV* y la *CD*, con el aporte de las Claves OTP de las autoridades electorales, las mismas se usarán en las etapas de preparación y cierre de la elección. En el transcurso del proceso debe garantizarse que las mismas se encuentren seguras y aisladas.
- Las claves K_{i1} aportadas por cada una de las *CA* autoridades servirán para dar valores a *ABV*. Inicialmente el mencionado archivo tendrá asignado cero en cada una de sus posiciones, luego se aplicará el XOR de cada una de las K_{i1} aportadas por las autoridades de la Junta electoral:

$$ABV = ABV \oplus K_{i1} \quad \forall i \quad 1 \leq i \leq CA \tag{1}$$

- Las claves K_{i2} de las autoridades inicializarán *CD*, que en un principio se encuentra con todos sus elementos en cero.

$$CD = CD \oplus K_{i2} \quad \forall i \quad 1 \leq i \leq CA \tag{2}$$

Etapa de Desarrollo de la Elección: se llevan a cabo las actividades que se detallan a continuación

- Autenticación del elector: consiste en verificar que quien va a votar figure dentro del padrón electoral, es decir que sea un votante válido. El modelo propone proceder de la forma habitual, registrándose el usuario en el lugar de la elección, con la presencia de una autoridad de la Junta Electoral. Para cada votante la autoridad habilitará un único sufragio.
- Emisión del voto: para el almacenamiento de los Votos se sigue el esquema MCDU.
- Para cada voto, el sistema genera una clave OTP (KV_v) de dimensión *TBslot* bits que tendrá un doble propósito:
- Aportar a la *CD* final de los votos, mediante operaciones XOR.
- Cifrar la información del voto.

Debe incluirse un protocolo que garantice que KV_v se mantenga inalterable para los dos usos en los que es necesaria.

El elector genera su voto que combinado con la clave produce la Contribución Final de Voto (*CFV*).

El aporte de la clave de voto a la *CD* se lleva a cabo a través de la siguiente operación:

$$CD = CD \oplus \mathrm{KV}_v \quad \forall v \quad 1 \leq v \leq N \tag{3}$$

Para cada Voto se genera la Cadena de Voto *CV*v formada por: *Id_de_Voto* (asignado aleatoriamente), *Id_de_ Cargo* y además el Id_Candidato seleccionado.

El sistema genera una cadena de TBslot bits CI_v, con todos sus elementos en cero. Produce además un conjunto de números aleatorios $CjtoQ = \{q_i\}$ para cada uno de los Q canales, donde qi representa el lugar donde se almacenará el voto en el canal i-ésimo.

Se realiza el XOR de la CV_v con los slots que corresponden a los q_i de la CI_v. Esto es:

$$Contribución_{vi} = CV_v \oplus CI_{vi} \quad \forall i \in CjtoQ \tag{4}$$

Finalmente se aplica:

$$CFV_v = Contribución_v \oplus KV_v \tag{5}$$

$$ABV = ABV \oplus CFV_v \tag{6}$$

Etapa de Cierre de la Elección y Recuento de Votos: Al momento de cierre de la elección se requiere la intervención de las Autoridades de la Junta Electoral.

El proceso de descifrado de los votos consta de tres sub-procesos:

- Aplicación de las K_{i1} de las Autoridades (las mismas que se usaron en la etapa inicial) a la última versión del ABV.

$$ABV = ABV \oplus Ki1 \; \forall i \;\; 1 \leq i \leq CA \tag{7}$$

- Aplicación de las K_{i2} de las Autoridades las mismas que se usaron en la etapa inicial)a la última versión de la CD.

$$CD = CD \oplus Ki2 \; \forall i \;\; 1 \leq i \leq CA \tag{8}$$

- XOR entre el ABV y la CD resultantes de los pasos anteriores que genera el Archivo Binario de Votos Descifrado ($ABVD$).

$$ABVD = ABV \oplus CD \tag{9}$$

- Finalmente, el recuento se realiza de la siguiente manera:
  - Se eliminan las tuplas que corresponden a votos vacíos.
  - Se eliminan las tuplas con votos que se generaron por colisiones y no se corresponden con la información de ninguno de los Id de voto.
  - Se recorre el $ABVD$ generándose la tabla de Votos Planos.
  - Luego se produce el conteo de los votos por medio de una consulta SQL.

3. Propuesta para verificabilidad "End to End"

Habiendo expuesto el funcionamiento detallado de OTP – Vote, se presenta a continuación una técnica que permite proveer al mismo de Verificabilidad *E2E*:

1. Se agrega un campo H a cada fila del archivo *ABV* de *OTP-Vote*. Nótese que será necesario modificar también el tamaño de *CD*.
2. Para cada sufragio, se genera un número aleatorio grande G, que se informa al votante y se hace público.
3. Se almacena en H el valor de una función *HASH(G)*.
4. Al finalizar el proceso se verifica que los números generados y publicados son coherentes con los valores almacenados en H.

Es necesario plantear una serie de condiciones iniciales que resultarán imprescindibles para garantizar la transparencia del proceso:

- Las claves K_{i1} y K_{i2} de la autoridad i-ésima se elegirán de una cantidad de claves mucho mayor que las que efectivamente se utilizarán. De ese grupo, serán seleccionadas de manera auténticamente aleatoria. El contenido de todas esas claves es información fundamental que debe ser protegida de manera incondicional.
- Cada clave incluirá una serie de commitments (que podrían colocarse en sobre cerrado). Por ejemplo, podría incluirse el resultado de h funciones de hash con $(h \in Z^{+}) \wedge (1 \leq h)$ aplicadas al contenido de dicha clave. Al finalizar el acto eleccionario, se publica cuales fueron las funciones utilizadas. Estas funciones serán similares para todas las claves de tipo K_{i1} y K_{i2}.
- El éxito del modelo exige que las siguientes informaciones críticas sean protegidas de manera inviolable:
  - El conjunto total de claves de tipo K_{i1}.
  - El subconjunto de claves de tipo K_{i1} que será efectivamente utilizado.
  - El conjunto total de claves de tipo K_{i2}.
  - El subconjunto de claves de tipo K_{i2} que será efectivamente utilizado.
  - El estado de los archivos *ABV* y *CD* luego de aplicar las claves K_{i1} y K_{i2}.

En esas condiciones, definimos:

Y: #Tuplas generadas.

N: #Votantes.

X_i: Valor obtenido en la i_ésima posición de la tupla.

C: #Componentes de la tupla. En todos los ejemplos posteriores se le asigna un valor 4, pero debe analizarse si otro valor es más conveniente.

Para ilustrar el funcionamiento del modelo propuesto, se implementa un procedimiento que, dado un valor de N, contabiliza cuántas de todas las tuplas posibles cumplen con la condición necesaria:

$$\sum_{i=1}^{C} X_i = N \qquad (10)$$

Luego, utilizando el software online SECANU[3] ([13]), por aplicación de los métodos de interpolación de Lagrange, se obtiene la fórmula para definir la cantidad de cuádruplas diferentes que pueden generarse en función de N:

$$Y = \frac{1}{6} N^3 + N^2 + \frac{11}{6} N + 1 \qquad (11)$$

Suponiendo que la función de hash elegida sea $H = G \bmod 4$, la Tabla 1 muestra la cantidad de tuplas posibles para diferentes valores de N.

N	*Y*
12	455
24	2925
36	9139
48	20825
60	39711
72	67525
84	105995
96	156849
108	221815
120	302621
132	400995
144	518665
250	2667126

Tabla 1: Variación de *Y* para diferentes valores de *N*.

Un valor de $Y = 250$ se aproxima a la cantidad de votantes promedio de una mesa electoral en la Argentina en la actualidad. Las cuádruplas no son equiprobables, pero todas tienen probabilidad mayor que cero.

A los efectos de analizar la validez de la propuesta, se realizó una simulación con 10.000.000 de corridas para elecciones con $N = 11$ votantes. Aplicando (11), se obtiene un valor de $Y = 364$ tuplas. En la Figura 4 se numeran las mismas desde 1 (11,0,0,0) hasta 364 (0,0,0,11) y se observa la distribución obtenida.

Es importante destacar algunos elementos que ocurrieron en la simulación:

- La totalidad de las tuplas posibles aparecieron en la simulación.
- Las que aparecen con mayor frecuencia son aquellas en los que los cuatro valores de la tupla (que sumados deben dar Y, en este caso 11) se aproximan a $N/4$.
- La suma de las 38 tuplas más frecuentes supera el 50% de las apariciones.
- La cuádrupla más frecuente (*CMF*) apareció 220.291 veces, es decir menos de 23 veces cada mil corridas.

[3] http://secanu.exactas.unlpam.edu.ar/

Parece lógico afirmar que si se aumenta el valor de N, se observará una disminución en la frecuencia de aparición de la *CMF*. Se implementa un nuevo simulador que contabiliza la cantidad de veces que aparece la misma, los resultados se muestran en la Tabla 2.

Cuando se realiza una sesión del simulador, es lógico que las cuádruplas más probables sean aquellas en las que todos los valores son cercanos al promedio. Esto se debe a la distribución uniforme de la variable aleatoria:

X_i="Valor de la función de Hash para el i_ésimo voto"

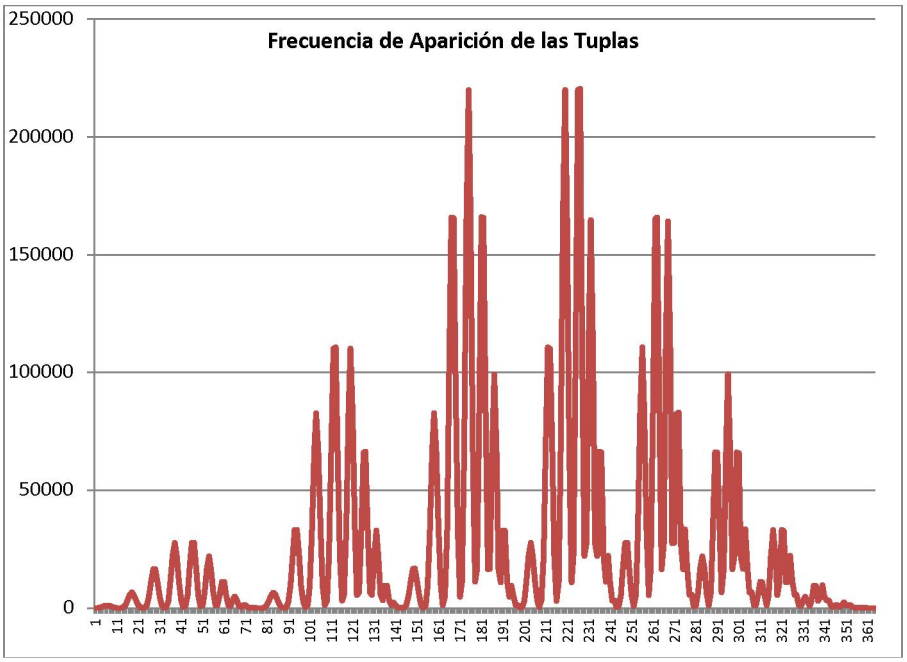

Figura 4: Frecuencia de Aparición de las Cuádruplas

N	*CMF*	F(*CMF*)	%	Repeticiones
5	2-1-1-1	5888	0,05888	100.000
10	3-3-2-2	2517	0,02517	100.000
15	4-3-4-4	1506	0,01506	100.000
20	5-5-5-5	1132	0,01132	100.000
25	6-6-7-6	787	0,00787	100.000
30	7-8-8-7	597	0,00597	100.000
35	9-9-9-8	460	0,0046	100.000
40	11-10-10-9	399	0,00399	100.000
45	11-12-11-11	343	0,00343	100.000

Tabla 2: Variación de los Valores de *CMF* en Función de *N*

Sin embargo, aún la *CMF* tiene un valor muy bajo. Que, además, se va reduciendo cuando aumenta el valor de *N*. Esto es promisorio. Para completar la evaluación de esta situación, en el simulador se realizan dos sesiones complementarias. En la primera se hace una sola sesión y se anota la Cuádrupla obtenida, (*CO*) y a continuación se realiza una corrida con 100.000 repeticiones y se observa en cuántas de ellas el resultado coincide con *CO*.Los resultados de las simulaciones se muestran en la Tabla 3. En ella, se define *TC* como el número total de cuádruplas posibles.

N	*CO*	*F(CO)*	*%*	*TC*
20	2-7-3-8	89	0,00089	1771
20	3-6-6-5	582	0,00582	1771
20	3-5-9-3	161	0,00161	1771
20	5-2-5-8	184	0,00184	1771
20	4-4-7-5	601	0,00601	1771
20	3-5-6-6	608	0,00608	1771
20	7-3-4-6	416	0,00406	1771
20	2-2-6-10	31	0,00031	1771
30	5-6-9-10	221	0,00221	5456
30	10-5-11-4	55	0,00550	5456
30	7-9-9-5	307	0,00307	5456
30	7-10-7-6	362	0,00362	5456

Tabla 3: Frecuencias de Aparición de Cuádruplas Específicas

4. Conclusiones y Trabajos Futuros

Se presenta en el presente documento una propuesta integral para proveer de verificabilidad *E2E* al sistema de voto electrónico OTP - Vote. La misma debe interpretarse de manera integral, dado que incluye una serie de componentes que proveen seguridad solamente si todos ellos se aplican de manera conjunta.

Desde el punto de vista práctico, la publicación del número aleatorio grande generado para cada sufragio, otorga la garantía al votante de que su voto fue tenido en cuenta. Simultáneamente, el chequeo de corrección de la tupla obtenida al final de la elección, contra la totalidad de los valores publicados da la seguridad de que la totalidad de los votos fue computada correctamente si se verifican las condiciones exigidas.

A futuro, es necesario hacer un profundo análisis sobre las funciones de hash que se aplicarán y sobre los parámetros correspondientes. Si bien aún la combinación más probable tiene una probabilidad muy baja, debe trabajarse para optimizar ese comportamiento. En el nivel ideal, todas las combinaciones serían equiprobables. En cualquier caso, debe procurarse minimizar la diferencia entre las combinaciones más y menos probables. El camino a seguir pasa por investigar el comportamiento de otras funciones de hash y, simultáneamente, verificar el comportamiento aumentando el valor de *C*. Si bien intuitivamente parece lógico que aumentar ese valor dará como resultado la obtención de mejores resultados, debe controlarse que no aparezcan "votos aislados" que permitirían probar por quién votó un elector específico. Ese

fenómeno se produce si todos los votantes que generaron el mismo valor de hash votaron por el mismo candidato. Obviamente, el riesgo de que eso ocurra aumenta cuando el valor de C es mayor.

Reconocimientos. A Jeroen van de Graaf, PhD., por su enorme generosidad para con el Mg. Pablo Garcia durante sus estadías en la Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Minas Gerais, Brasil, en 2012 y 2016.

Referencias

1. Ryan P., Schneider S., Teague V.: "End-to-End Verifiability in Voting Systems, from Theory to Practice". Voting Systems, from Theory to Practice. IEEE Security & Privacy, 13(3):59–62, 2015.
2. Rabin, M., Rivest, R.: "Efficient End to End Verifiable Electronic Voting Employing Split Value Representations" Bregenz, Austria. EVOTE 2014. ISBN 978-9949-23-688-6.
3. Awad M., Leiss E.: "End-to-End Cryptography: Spreading Democracy". International Journal of Applied Engineering Research. Volume 11, Issue 11. Ps. 7391-7394. 2016.
4. Bast, S. "Confidencialidad e Integridad de Datos en Sistemas de E-Voting – Un Modelo para la Implementación Segura de un sistema de Voto Presencial" - Editorial Académica Española.- ISBN 978-3-639-53793-2. 2017
5. Van de Graaf J.: "Anonymous One Time Broadcast Using Non Interactive Dining Cryptographer Nets with Applications to Voting". Towards Trustworthy Elections". Ps. 231-241. Springer-Verlag Berlin, Heidelberg. ISBN:978-3-642-12979-7. 2010.
6. Bast S., Montejano G., García P., Fritz E.: "Evaluación de la Integridad de Datos en Sistemas de e-Voting". XVII Workshop de Investigadores en Ciencias de la Computación (WICC 2015). 2015. Universidad Nacional de Salta. Ps. 827 -831. ISBN: 978-987-633-134-0. http://sedici.unlp.edu.ar/handle/10915/46871.
7. N. Nagaraj, V. Vaidya, and P. G. Vaidya, "Revisiting the one-time pad," International Journal of Network Security, vol. 6, no. 1, pp. 94-102, 2008.
8. Shannon, C. E.: "Communication Theory of Secrecy Systems"- Bell System Technical Journal - N° 28 (1949) 656–715.
9. García, P.: "Una Optimización para el Protocolo Non Interactive Dining Cryptographers" - ISBN-13: 978-3-639-85270-7. ISBN-10: 3639852702. EAN: 9783639852707. Editorial Académica Española. https://www.eae-publishing.com/ - 2017.
10. García, P., van de Graaf J., Hevia A., Viola A.: "Beating the Birthday Paradox in Dining Cryptographers Networks". En "Progress in Cryptology – Latincrypt 2014". Springer International Publishing. ISSN: 0302-9743. ISSN (electrónico): 1611-3349. ISBN: 978-3-319-16294-2. ISBN (eBook): 978-3-319-16295-9. Ps. 179 – 198. Octubre, 2014.
11. García P., van de Graaf J., Montejano G., Riesco D., Debnath N., Bast S.: "Storage Optimization for Non - Interactive Dining Cryptographers (NIDC)". The International Conference on Information Technology: New Generations. 2015. Las Vegas, Nevada, USA. http://ieeexplore.ieee.org/document/7113449/.
12. García P., Bast S., Fritz E., Montejano G., Riesco D., Debnath N.: "A Systematic Method for Choosing Optimal Parameters for Storage in Parallel Channels of Slots". IEEE International Conference on Industrial Technology (ICIT 2016). 14 - 17 March 2016 / Taiwan, Taipei. http://ieeexplore.ieee.org/document/7475019/.
13. Ascheri M., Pizarro R., Astudillo G., García P., Culla M., Pauletti C.: "Software Educativo para la Resolución Numérica y Gráfica de Temas de Cálculo Numérico". WICC 2017. Instituto Tecnológico Buenos Aires. 2017.